# Proposal & Costing Review

## WBS Level 2 System Summaries

## *for*

## *BTeV Offline*

**May 15, 2000**

# Table of Contents

# 1. Analysis and Simulation Facilities

**General Description**

The simulations and analyses needed to produce physics results from the BTeV data will be carried out on a farm of PCs. The costing of this section is based on the assembly, coding, and operation of the PC farms recently built for CDF and D0. The BTeV needs are based on the experience gained from producing the physics simulations incorporated in this BTeV proposal on the new CDF farm.

**Summary**

The costs of the Simulation and Analysis Facilities include, in summary: $1.3 for analysis computing, $2.9M for data archiving, and $600K for desktop systems. The costs total $4.4M for materials, $250K for labor, and $610K (or 13%) contingency.

# 2. Algorithm and Other Software and Integration

**General Description**

The item refers to the software we need to write in order to translate a raw stream of input data from the detector to a set of reconstructed HEP event samples. Unlike many other HEP experiments, the offline and online software systems are closely related. In fact, algorithms and codes will be shared. For instance, the Level 1 and Level 2 trigger algorithms (See PTDR'99) operate on a perfectly aligned detector. The charged particle tracker must be aligned before one can take data. Alignment - or alignment checking - software will have to run in the early part of any high performance data run. Thus this software must be ready prior to data taking.

The second reason for early development of reconstruction software is strictly economic: the cost of computer memory, CPU, and related network gear is decreasing faster than the cost of high performance, high volume magnetic storage systems. Level 3 and definitely Level 4 (strictly offline) will be mostly data compression. Hopefully, we will keep in long-term (years) storage only the final reconstructed data, consisting of a set of vertices and particles, with limited information on the reconstruction quality of the associated tracks or neutral clusters (photons and $\pi^0$)

This strategy implies significant risks: HEP data reconstruction software is often not very robust, and, even if perfectly written, requires fine tuning based on real data rather than Monte-Carlo. However, we feel confident that recent progress in simulation and track fitting, backed with high performance farms will allow us to minimize the amount of time required to tune the reconstruction code. Note that recent HEP experiments (CDF Run I, BABAR, *etc.*) are indeed able to reconstruct their data in record time (days, maximum a few weeks) after data taking. We plan to aggressively follow this trend.

To analyze data in almost real time, the software must be very carefully written. The software must, and will, be properly engineered from the start. Software engineers and professional programmers will be involved in this task. Even for prototyping, or preliminary algorithm searches, we rely on computer professionals for system management and application software installation and support. More importantly, software engineers advise us on how to layout C++ classes or other data structures, right from the start. While their involvement will be small at the onset, final design and implementation of these software modules should be done in close collaboration with software engineers and dedicated programmers. This includes software reviews and software management.

Complicated algorithms will need to be rewritten (more than once) as we gain knowledge about the BTeV spectrometer performance. In particular, background and related detector occupancies can't be exactly derived from simulation. Thus, algorithm modifications must be planned and budgeted ahead of time. Diagnostic tools and monitoring will play an essential role.

Note also that we must plan code migration away from FORTRAN in an orderly fashion, and be ready for changes in operating system or compilers, debuggers, linkers, memory leak detectors and so forth.

**Major Component Subsystems**

*Algorithm and Design Studies*: Based solely on Monte-Carlo data (MCFast and BTeV Geant), triggers and reconstruction code will be prototyped. Feasibility studies for a given final state must be determined, based on partially reconstructed data.

*Reconstruction Code, System Development, Implementation, and Testing*: This item is the main cost driver of the BTeV software. This item includes the Monte-Carlo packages, the triggers, the complete reconstruction code, Heavy Quark final state classification, the alignment system, final offline filtering and "skimming" engines (selection of sub-samples based on track or vertex information). Finally we mention "Analysis", i.e., studies of Dalitz decay, mixing via lifetime plots, corrections for acceptance of branching fractions and so forth.

*Tools*: Such a large software system cannot be constructed and analyzed without an appropriate toolbox. This includes not only good compilers, but also data bases, data browsers, plotting packages and so forth. We plan to use "freeware" or "shareware" as much as possible. The possible exceptions are debuggers and memory analyzers, 3D graphics and final data presentation tools. Such tools are relatively cheap (the expensive ones are highly specialized) and not all BTeV members need to access them.

*Software Code Management*. Given the critical nature of this software, it must be managed carefully. Tools and local Fermilab expertise will be used.

Cost estimates are not very well known at this point. Labor costs are by far the biggest component. Note that there is great uncertainty in estimating the amount of code that needs to be written. We have listed projects that have already started, if not partially completed. We have also had informal conversation with our CDF and D0 colleagues involved in Run II software. However, given the uncertainties listed above, a contingency of about 50% is used. Clearly, this cost estimate, and this document, is preliminary. The process of writing this software is also iterative in nature, hence, more accurate versions of this document will be possible as prototypes are written, along with the ad-hoc engineering data.

**Summary**

The costs of the Algorithms and Integration Software include, in summary: $2.9M for algorithm and code development, $3.5M for software tools and code management. The costs total $4.7M for labor, and $2.4M (or 50%) contingency.